



2.0 Approach

At iCubicles, we focus on minimizing business and technical risks. This is so called Risk-Down model based on over 10 years of experience in outsourcing and offshore software development.

It's no secret where those risks lay - project management, communication, information exchange, documenting and handling business side of projects.

Risk-Down Model

How it usually goes: in 80% of cases, customer comes with a draft vision of their needs. It may be good and prospective, but it's only an idea. Usually it sounds like: "Guys, I think I can clone YouTube, mix it with Kaboodle, add 2 features and it will bring me millions".

Unless it's Einstein's " $e=mc^2$ " level, the idea itself is not enough. If not successful, a project won't bring our customer ROI and leave us without satisfaction in any way. That's why we don't go further until our marketing analysts got a clear picture of market, competitors, their pros and cons, sensible niche and positioning, revenue channels and pricing. At this stage technical risks will be tested by technology experts of iCubicles.

In the next iteration, our usability experts and GUI designers develop proper communication routes with project audience. We work out interfaces and usability concept, describe user scenarios, flows, use cases, and build concept design. In the end, all that is integrated into an all-inclusive Software Requirements Specification (SRS) and provides a basement for further technical development.

Before the production, we convert high level requirements into an executable production plan. Our goal here is to elaborate the initial idea further into a substance that can be conditionally "easily" deployed at any development center. With more efforts of software architects and analysts we transform Business Vision and SRS into a complete and consistent Software Architecture and Development Specification.

At production stage we represent customer interests, playing as a undisputable authorized quality control and compliance managers. Project visibility is supported, for example, by weekly progress reviews of what is done last week and going to be done this week; problems and achievements; hours spent for planned tasks; updated source code delivery and so on.

In the following table you will see an overview of risk factors, possible outcome and actions we take to decrease those possibilities.

Factor	Potential Risks	iCubicles Risk-Down Solution
Raw business model	<ul style="list-style-type: none"> - Money/time waste - Poor market chances - Lack of perspectives 	Feasible business model – market review, competition analysis, positioning, consistent revenue model, usability expertise
Lack of prototyping	Stoppage of project at later stages due to technical risks impact	Mandatory technical prototyping in case of innovative projects
Inadequate transfer of business model into Software Requirements	<ul style="list-style-type: none"> - Loss of important details - Incorrect flows and scenarios - Time and budget overrun 	<ul style="list-style-type: none"> - Consistency checking - Control of business analysts - Multi-iterated approach
Inconsistent Technical Specifications	<ul style="list-style-type: none"> - Time plan breakdown - Budget overrun - Painful extra negotiations - Deviation detection at later project stages - Budget waste - Project failure 	<ul style="list-style-type: none"> - Consistency checking - Empty Realization of Classes - Deep pre-project analysis - Solid Technical Specification standards - Possible project prototyping
Offshore development	<ul style="list-style-type: none"> - Miscommunication - Hard transfer of industry knowledge - Allocation of lower level resources at high cost - Lower influence over a team 	<ul style="list-style-type: none"> - Mandatory screening of project team (checking of certificates, resumes, comprehension checklist, effectiveness rating) - Regular control of initial allocation conditions - Intermediate communication
Low Production Visibility	<ul style="list-style-type: none"> - Loss of project direction - Hijacking of source codes - Inadequate allocation - Insufficient change tracking and loss of details - Incomplete versions - Time/budget overrun - Project failure 	<ul style="list-style-type: none"> - Monitoring of business side support team - Weekly project status reviews - Detailed hourly reports - Achievements and problem analysis - Access to updated source codes - Early problem detection
Poor QA practices	<ul style="list-style-type: none"> - Time plan breakdown - Loss of market reputation - Budget overrun 	<ul style="list-style-type: none"> -Independent QA analysis -Business compliance revision -Solid QA plan standards -Acceptance procedure

Team Dedication

You will have full-time, exclusive, dedicated team guided by our analysts and team leader(s). And they are going to work literally 24x7 in a project, think about it round-the-clock. Pretty much like we work on our own software projects. We do believe very strongly that this is the only way for a success of any business-oriented project. Also for the most crucial project phases we can organize a team to work onsite, close to customer or at their office. This model, with Team Leader(s) at customer end and the rest team off-site would give the top level of reliability and fast progress while still keep the cost of a lower level. At the same time we build so detailed production documents and provide complete project

management, that actually the coding side itself can be done anywhere by experienced developers, with little to no extra questions and guidance. You would keep high level of flexibility in this matter.

It does not make any sense to describe all the technologies we are possessing. You can be sure that any technical skill will be supplied on a high level from within our in-house team or under our absolute control from a certified provider.

Below you will find a description of our standard project workflow and average timeline for different phases.

General Project Workflow

Project Development Phases (starting from a draft business vision, idea in general)		Timeline
Technical Side	Business Side	
Technical Analysis: <ul style="list-style-type: none"> - investigation of competitors - researching patents and patent pending technologies - innovation search - draft of possible alternates and risks - development specifics, pros and cons - prototyping plan 	Business Overview: <ul style="list-style-type: none"> - money making capabilities - competition overview - affiliate/ads/paid services opportunities - positioning ideas - market screening and trends - potential risks (commercial, juridical) - clarification questions 	1 week
Technical Prototyping: <ul style="list-style-type: none"> - testing of expected technical risks - basic technical model for further development - prototyping of data structures and architecture 	Business Vision: <ul style="list-style-type: none"> - main direction - competition analysis - unique value positioning - service structure and features - roles, users, stockholders - user motivation - distribution model - revenue channels - "good-enough-to-start" and long-term growing plan 	1-2 weeks
	++ Extra Services <ul style="list-style-type: none"> - Naming - Graphic Design - Corporate Identity - Deep Market Research 	1-3 weeks 2-4 weeks 1-2 weeks 2-4 weeks
Software Requirements Specification (explicitly understood, consistent, complete, relevant, optimally detailed, and convenient to comprehend) including: <ul style="list-style-type: none"> - Scenarios and Flows for system users <ul style="list-style-type: none"> - Valid Usability Model - Design Concept, Style Guide - Comprehension Checklist (for development team) 		2-3 weeks

System Architecture: <ul style="list-style-type: none"> - Modules and Interfaces - Technical Requirements - Functional Requirements - System Requirements - Architecture alternates - Client-side and Server-side specifics 		1-2 weeks
Project Initiation Seminars (can substitute complete technical specification in agile development or time aggressive constraints)		2-3 times during 2 weeks
Technical Specification and Production Plan (explicitly understood, consistent, complete, relevant, optimally detailed, and convenient to comprehend)		1-2 weeks
Technical Specs Revision including Empty Implementation (of structure and classes) and Consistency Verification		1-2 weeks
Project Planning <ul style="list-style-type: none"> - Overall production layout - Mid term milestones - Short term work slots and task decomposition - Time, Resources allocation table - Reporting standards Quality Assurance <ul style="list-style-type: none"> - Unit testing - Milestones QA plans - Version Revisions - Final Testing and Bugfixing 	Project Tracking <ul style="list-style-type: none"> - Day to day communication and review - Compliance monitoring - Possible deviations analysis - Effectiveness monitoring <i>Weekly Progress Reviews:</i> <ul style="list-style-type: none"> - Detailed hourly reports - Last week achievements and problems - Forthcoming week plan - Up-to-date source codes/ Repository access - User instructions - Overall progress report 	During the production phase
Fine Tuning and Delivery	Final Compliance Revision and Acceptance	1-2 weeks
Project Release		
Post-Release Maintenance		
Maintenance and Support	Project Growth Plan Implementation	
	Market Traction and Analysis	
	Project Change Planning	